IN THE CLAIMS:

Please amend the claims as follows:

Claims 1 and 53 (Cancelled)

54. (Currently Amended) A file system that stores and retrieves information in ~~,~~ ~~having~~ a plurality of storage blocks, said file system ~~and~~ including a plurality of bits associated with each one of said plurality of storage blocks, at least one of said plurality of bits identifying whether said one storage block was part of said file system at a time earlier than a current consistent version of said file system, and at least a second one of said plurality of bits identifying whether said one storage block was part of said file system at a second time earlier than a current consistent version of said file system.

55. (Cancelled)

56. (Currently Amended) A file system as in claim 54 ~~55~~, including an element disposed for selecting storage blocks in response to said one bit and said second one bit associated with said selected storage blocks.

57. (Previously Presented) A file system as in claim 56, including an element disposed for copying said selected storage blocks to a destination.

58. (Previously Presented) A file system as in claim 57, wherein said destination includes: a tape, a disk, a data structure in a second file system, a set of network messages, or a destination distributed over a plurality of file systems.

59. (Previously Presented) A file system as in claim 54, including an element disposed for selecting storage blocks in response to said one bit associated with said selected storage blocks.

60. (Previously Presented) A file system as in claim 59, including an element disposed for copying said selected storage blocks to a destination.

61. (Previously Presented) A file system as in claim 60, wherein said destination includes: a tape, a disk, a data structure in a second file system, a set of network messages, or a destination distributed over a plurality of file systems.

62. (Currently Amended) A file system having a plurality of storage blocks, said file system including a snapshot including a set of member storage blocks selected from said plurality, said member storage blocks forming a consistent file system other than an active file system, ; said

snapshot being disposed as an object in said file system ~~, wherein said file system is responsive to at least one file system request with regard to said snapshot~~.

63. (Previously Presented) A file system as in claim 62, including

a mark on allocate image of a set of member storage blocks selected from said plurality, said member storage blocks having been added to said snapshot; and

a storage image defined based on said snapshot and said mark on allocate image, said storage image indicating a set of member storage blocks selected from said plurality.

64. (Previously Presented) A file system as in claim 63, wherein said storage image is defined with regard to a logical sum operation on said snapshot and said mark on allocate image.

65. (Previously Presented) A file system as in claim 62, including

a mark on deallocate image of a set of member storage blocks selected from said plurality, said member storage blocks having been removed from said snapshot; and

a storage image defined based on said snapshot and said mark on deallocate image, said storage image indicating a set of member storage blocks selected from said plurality.

66. (Previously Presented) A file system as in claim 62, including

a shadow snapshot of a set of member storage blocks selected from said plurality, said member storage blocks having formed a consistent file system other than an active file system, with a set of selected member storage blocks removed from said consistent file system; and

a storage image defined based on said snapshot and said shadow snapshot, said storage image indicating a set of member storage blocks selected from said plurality.

67. (Previously Presented) A file system as in claim 62, including an indicator of which ones of said member storage blocks have been copied.

68. (Previously Presented) A file system as in claim 62, including a plurality of said snapshots; wherein said plurality of said snapshots are associated with an array of bits, said array having one set of bits for each storage block in said plurality of storage blocks, said set of bits having at least one bit for each said snapshot.

69. (Previously Presented) A file system as in claim 62, wherein said file system can manipulate said snapshot without having to traverse a hierarchy of file system objects within said snapshot.

70. (Previously Presented) A file system as in claim 62, wherein said snapshot includes a data structure disposed in a format allowing for a set management operation to be performed efficiently.

71. (Previously Presented) A file system as in claim 62, wherein said snapshot includes an array of bits, said array having one bit for each storage block in said plurality.

72. (Previously Presented) A file system as in claim 62, including

a plurality of said snapshots; and

a storage image determined based on said plurality of snapshots;

said storage image defining a second set of member storage blocks selected from said plurality.

73. (Previously Presented) A file system as in claim 72, including an indicator of which ones of said storage blocks in said storage image have been copied.

74. (Previously Presented) A file system as in claim 72, wherein said storage image is a result of a logical sum or difference performed on said set of member storage blocks for said snapshot and a set of member storage blocks for a second said snapshot.

75. (Previously Presented) A file system as in claim 72, wherein said storage image is a result of a logical sum or difference performed on said set of member storage blocks for said snapshot and a set of member storage blocks for a second said storage image.

76. (Previously Presented) A file system as in claim 72, wherein said storage image is a result of a set management operation on said set of member storage blocks for said snapshot.

77. (Previously Presented) A file system as in claim 62, wherein said snapshot includes a data structure disposed in a format allowing for a set management operation to be performed in O(n) time or less, where n is a number of storage blocks in said plurality, without reading any contents of said storage blocks in said plurality.

78. (Previously Presented) A file system as in claim 77, wherein said set management operation is a logical sum or difference.

79. (Previously Presented) A file system as in claim 62, wherein said snapshot includes a data structure identifying which storage blocks in said plurality are member storage blocks of said snapshot.

80. (Previously Presented) A file system as in claim 79, wherein said data structure uses no more than 1/100th of an amount of storage required by said storage blocks in said plurality.

81. (Previously Presented) A file system as in claim 79, wherein said data structure uses no more than four bytes per storage block in said plurality.

82. (Previously Presented) A method of operating a file server, said method including steps for

forming a first snapshot of a first consistent state of said file system at a selected time, said first snapshot including an indication of a set of storage blocks in said first consistent state;

forming a second snapshot of a second consistent state of said file system, said second snapshot including an indication of a set of storage blocks in said second consistent state; and

performing an operation on said first and second snapshots to form a storage image including an indication of at least some storage blocks in said file system.

83. (Previously Presented) A method as in claim 82, wherein said operation includes a logical sum or difference.

84. (Previously Presented) A method as in claim 82, wherein

said operation includes a logical sum or difference; and

a purpose of said operation includes making a copy including or excluding a selected range of snapshots.

85. (Previously Presented) A method as in claim 82, wherein

said operation includes a logical sum or difference; and

a purpose of said operation includes copying said storage image to a destination.

86. (Previously Presented) A method as in claim 85, wherein said destination includes a tape, a disk, a data structure in a second file system, a set of network messages, or a destination distributed over a plurality of file systems.

87. (Currently Amended) A method to be performed in a file system, said file system having a plurality of storage blocks, said method including steps for

defining a storage image of a set of member storage blocks selected from said plurality, said storage image being formed based on a set of member storage blocks forming a consistent file system other than an active file system; ~~and~~

forming an image stream of a sequence of member storage blocks selected from said storage image; and

sending said image stream from a source file system to a destination file system.

88. (Previously Presented) A method as in claim 87, including steps for associating a block location with each one of said sequence.

89. (Previously Presented) A method as in claim 87, further including steps for reconstructing a file system based on said image stream.

90. (Previously Presented) A method as in claim 87, wherein said steps for forming are performed in response to a selected operation to be performed on said member storage blocks, said selected operation being other than an operation on an active file system

91. (Previously Presented) A method as in claim 87, wherein said steps for forming include steps for optimizing said sequence of member storage blocks for a file system operation.

92. (Previously Presented) A method as in claim 87, wherein said steps for forming include steps for optimizing said sequence of member storage blocks for a file system operation in a RAID file system.

93. (Previously Presented) A method as in claim 87, wherein said steps for forming include steps for

optimizing said sequence of member storage blocks based on a physical location in a storage medium for each said member storage block, said storage medium having a plurality of storage elements capable of being read in parallel; and

ordering said sequence of member storage blocks so that said member storage blocks can be optimally read in parallel from said plurality of storage elements.

94. (Previously Presented) A method as in claim 87, wherein said storage image represents a complete file system.

95. (Previously Presented) A method as in claim 87, wherein said storage image represents a set of changes to a file system.

96. (Previously Presented) A method as in claim 87, including repeating said defining step at periodic intervals.

97. (Previously Presented) A method as in claim 87, including repeating said defining step in response to an operator command.

98. (Previously Presented) A method as in claim 87, including repeating said selecting step in response to a remote device.

Claims 99 to 103 (Cancelled)

104. (Currently Amended) Apparatus including a file system that stores and retrieves information, said file system including a plurality of snapshots thereof, each representing an associated consistent state at an associated selected time; and each said snapshot including an indication of a set of storage blocks in said associated consistent state, said indication being recorded in at least one storage block in said associated consistent state.

105. (Previously Presented) Apparatus as in claim 104, including a storage image defining at least some storage blocks in said file system, said storage image based on an operation on at least two of said snapshots.

Claims 106 to 110 (Cancelled)

111. (Previously Presented) In a file system having a plurality of storage blocks, a data structure including

a first snapshot of a set of member storage blocks selected from said plurality, said member storage blocks forming a consistent file system other than an active file system;

said first snapshot being represented as an object in said file system and having a set of storage blocks for recording said first snapshot;

whereby copying said member storage blocks in said first snapshot has the property of preserving at least one snapshot recorded in said file system at a time of said first snapshot.

112. (Previously Presented) A data structure as in claim 111, including

a second snapshot of a set of member storage blocks selected from said plurality, said member storage blocks forming a consistent file system other than an active file system;

said second snapshot being represented as an object in said file system and having a set of storage blocks for recording said second snapshot;

whereby copying said member storage blocks in said second snapshot has the property of preserving at least one snapshot recorded in said file system at a time of said second snapshot.

113. (Previously Presented) A data structure as in claim 111, including

an image stream including a set of storage blocks including both said first snapshot and said second snapshot;

whereby copying said member storage blocks in said image stream has the property of preserving both said first snapshot and said second snapshot.

114. (Previously Presented) In a file system having a plurality of storage blocks, a data structure including

a snapshot of a set of member storage blocks selected from said plurality, said member storage blocks forming a consistent file system other than an active file system;

said snapshot being represented as an object in said file system and having a set of storage blocks for recording said snapshot;

whereby a backup and restore operation on said file system has the property of preserving said snapshot within said file system.

Claims 115 to 120 (Cancelled)

121. (Currently Amended) A ~~In a~~ file system that stores and retrieves information in ~~having~~ a plurality of storage blocks, including a data structure ~~stored in said file system,~~ including a shadow snapshot of a set of member storage blocks selected from said plurality, said member storage blocks having formed a consistent file system other than an active file system, with a set of selected member storage blocks removed from said consistent file system;

wherein said shadow snapshot is disposed as a single object in said file system, whereby said file system can manipulate said snapshot without having to traverse a hierarchy of file system objects within said snapshot.

122. (Currently Amended) A file system ~~data structure~~ as in claim 121, wherein said shadow snapshot is disposed in a format allowing for a set management operation to be performed efficiently.

123. (Currently Amended) A file system ~~data structure~~ as in claim 121, wherein said shadow snapshot uses, in addition to said member storage blocks, no more than 1/100th of an amount of storage required by said storage blocks in said plurality.

124. (Currently Amended) A file system ~~data structure~~ as in claim 121, wherein said shadow snapshot uses, in addition to said member storage blocks, no more than one byte per storage block in said plurality.

125. (Cancelled)

126. (Currently Amended) A file system ~~data structure~~ as in claim 121, wherein said removed member storage blocks are responsive to completion of a processing operation.

127. (Currently Amended) A file system ~~data structure~~ as in claim 126, wherein said processing operation includes a file system operation.

128. (Currently Amended) A file system ~~data structure~~ as in claim 126, wherein said processing operation includes reuse of said selected member storage blocks by said file system.

129. (Currently Amended) A file system ~~data structure~~ as in claim 121, wherein said shadow snapshot is disposed in a format allowing for a set management operation to be performed in O(n) time or less, where n is a number of storage blocks in said plurality, without reading any contents of said storage blocks in said plurality.

130. (Currently Amended) A file system ~~data structure~~ as in claim 129, wherein said set management operation is a logical sum or difference.

131. (Currently Amended) A ~~In a~~ file system that stores and retrieve information in ~~having~~ a plurality of storage blocks ~~,~~ including a data structure that includes ~~including~~ a mark on

allocate image of a set of member storage blocks selected from said plurality, said member storage blocks having been added to a snapshot that originally formed a consistent file system;

wherein said mark on allocate storage image is disposed as a single object in said file system, whereby said file system can manipulate said snapshot without having to traverse a hierarchy of file system objects within said snapshot.

132. (Cancelled)

133. (Currently Amended) A file system ~~data structure~~ as in claim 131, wherein said mark on allocate image is disposed in a format allowing for a set management operation to be performed efficiently.

134. (Currently Amended) A file system ~~data structure~~ as in claim 131, wherein said mark on allocate storage image uses no more than 1/100th of an amount of storage required by said storage blocks in said plurality.

135. (Currently Amended) A file system ~~data structure~~ as in claim 131, wherein said mark on allocate image uses no more than four bytes per storage block in said plurality.

136. (Currently Amended) A file system ~~data structure~~ as in claim 131, said member storage blocks having been selected responsive to completion of a processing operation.

137. (Currently Amended) A file system ~~data structure~~ as in claim 136, wherein said processing operation includes a file system operation.

138. (Currently Amended) A file system ~~data structure~~ as in claim 136, wherein said processing operation includes reuse of said selected member storage blocks by said file system.

139. (Currently Amended) A file system ~~data structure~~ as in claim 131, wherein said mark on allocate image is disposed in a format allowing for a set management operation to be performed in O(n) time or less, where n is a number of storage blocks in said plurality, without reading any contents of said storage blocks in said plurality.

140. (Currently Amended) A file system ~~data structure~~ as in claim 139, wherein said set management operation is a logical sum or difference.

141. (Currently Amended) A ~~In a~~ file system that stores and retrieve information in ~~having~~ a plurality of storage blocks ~~,~~ including a data structure that includes ~~stored in said file system, including~~ a mark on deallocate image of a set of member storage blocks selected from said plurality, said member storage blocks having been removed from a snapshot that originally formed a consistent file system;

wherein said mark on deallocate storage image is disposed as a single object in said file system, whereby said file system can manipulate said snapshot without having to traverse a hierarchy of file system objects within said snapshot.

142 (Cancelled)

143. (Currently Amended) A file system ~~data structure~~ as in claim 141, wherein said mark on deallocate image is disposed in a format allowing for a set management operation to be performed efficiently.

144. (Currently Amended) A file system ~~data structure~~ as in claim 141, wherein said mark on deallocate storage image uses no more than 1/100th of an amount of storage required by said storage blocks in said plurality.

145. (Currently Amended) A file system ~~data structure~~ as in claim 141, wherein said mark on deallocate image uses no more than four bytes per storage block in said plurality.

146. (Currently Amended) A file system ~~data structure~~ as in claim 141, wherein said mark on deallocate image is disposed in a format allowing for a set management operation to be performed in O(n) time or less, where n is a number of storage blocks in said plurality, without reading any contents of said storage blocks in said plurality.

147. (Currently Amended) A file system ~~data structure~~ as in claim 146, wherein said set management operation is a logical sum or difference.